blue rabbit
sustainable product strategies

research

Cameras and Vision
Conceptual Possibilities
for Mobile Robots

introduction

This document is an extremely simplistic introduction to the concepts of cameras applied to machine vision. The simplistic nature of this document means that it should not be taken as providing definitive answers, it is intended as a high level guidance tool to assist in decision making at that level. Technological details and actual hardware and software configurations belong at the lower level of a technical working group.

This document also looks specifically at cameras, in the popular sense that ordinary people imagine when they hear the term. That is to say, a camera is an instrument that captures images (either individually or in timed sequences) using those wavelengths of the electro-magnetic spectrum that are visible to the human eye. This visible spectrum may be extended slightly into both the infrared and the ultraviolet, but not far enough to become (say) microwaves.

Machine vision can be purely camera based, or may be enhanced with sensors that work in different bands of the electro-magnetic spectrum, such as infrared sensors, RADAR, LIDAR, X-rays, and so on. This document focuses solely on visible light optical cameras, with the intention of understanding the issues associated with mounting one or more such cameras on a mobile robot platform (a mechatron) and doing something useful with the image sets that they can yield.

Ultimately the issue comes down to the question of "What's it for?" There are many, many possibilities, but each potential application carries a significant cost in the software and processing horsepower needed to realise it. Worse, at the current state of the art (not market) each application is typically isolated, with startlingly little in common, other than some shared libraries of primitives. This means that there are effectively no savings when camera-derived data streams are put to more than one use, and there will be all the associated resource issues for each additional application. The current state of the market requires that each application have one or more dedicated cameras associated with it.

Onboard Digital Camera System Overview

Figure 1 gives a highly simplified view of the major components involved with a digital camera system mounted on a mechatron. Not all the components will necessarily be present in a single camera installation. Equally, there may be multiple instances of some or all of the components in a single mechatron, depending on the design.

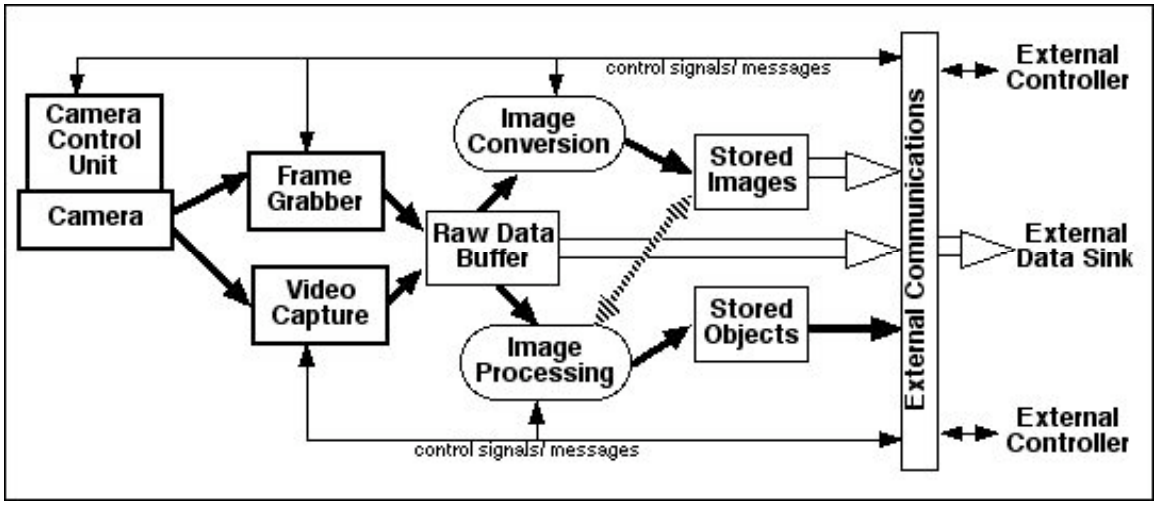

Figure 1: Conceptual Components of an Onboard Digital Camera System.

This picture changes radically if an analogue video camera replaces the digital camera: see Figure 2, and the discussion in Section 2.

The major conceptual components of an onboard digital camera system are:

Digital Camera

Camera Control Unit (CCU)

Frame Grabber

Video Capture

Raw Data Buffer

Digital Image Conversion and Compression

Digital Image Store

Digital Image Processing

Object Store

Control and Data Busses

External Communications

The following sections describe each of these conceptual components to a level of detail that should be sufficient to enable their relative importance and salient characteristics to be evaluated.

Digital Camera

The digital camera is the piece of technology that everyone recognizes, even if these days cameras no longer look like the still and video cameras of yore. The most important optical component is the lens, or lenses, followed by the mechanisms for setting focal length, aperture, shutter speed, and so on (or, at least, their digital equivalents). The

camera may have target illumination capabilities as an integral component (think flash or floodlight) or these may be external to the camera system, albeit co-located on the same mechatron.

In the world of digital "photography" there is little physical difference between still and video cameras. In consumer packages the differences all reflect the increased processing and storage required for acquiring video, as well as the need for continuous rather than flash illumination. A digital video camera can be thought of as a still camera taking a continuous series of still images (frames) at a pre-determined frame rate.

The most significant issue for an onboard camera is whether it is mounted in a fixed position, thus having a fixed field of view, or is installed in some kind of movable mounting. Movable mountings may be single axis, giving either side to side swivel (pan) or up and down (tilt) capability, or may provide multi-axis capabilities. Movable camera mounts require a programmable control unit, very probably with remote control via external (off-board) telemetry.

Finally, cameras fitted to mechatrons need to be ruggedized to a degree that matches the operational envelope of the mechatron on which they are mounted. For instance, a simple (and cheap) webcam is designed to be stationary at all times, with extremely intermittent gentle adjustments of position. Mounting it on even an indoor robot will subject it to vibration and the shocks of encounter when the mechatron comes into contact with a solid object, really necessitating some form of isolation mounting. The ultimate degree of mounting sophistication is to have an auto-stabilising mounting (like a movie Steadicam) that attempts to keep the camera's point of view stable no matter what gyrations the carrier of the whole system (the mechatron) performs, The sophistication (and cost) of all these has to be scaled up for all-weather, all-terrain work in the great outdoors, possibly on a mechatron travelling at high speeds even over rough country.

Camera Control Unit (CCU)

The camera control unit (or CCU) is the electronic or electro-mechanical device that controls all the operations of the camera necessary to perform photography. For a simple webcam mounted in a fixed housing the controls will be extremely simple. The more sophisticated the camera and its mounting, the more complex the CCU. The CCU may be program controllable from an onboard computer running suitable software, but is very likely to have a remote control capability, either instead of, or in addition to, local automated control.

Typical controls include pan, tilt, zoom, lens selection, aperture setting, target illumination, view port cleaning (wipers, etc.), focus, frame rate, and so on. The control unit also has to be ruggedized to match the intended operating environment of the mechatron on which it is mounted.

Frame Grabber

Conceptually, the frame grabber is the piece of the system that snaps a photograph, committing it to a raw data buffer in the native image format of the device. This raw image will reflect the physical characteristics of the imaging component of the camera, and is typically quite large, the data size increasing with the square of the resolution expressed as number of pixels, and proportionally with the number of layers of sense data per pixel. Raw frame data sizes are typically measured in megabytes rather than kilobytes. These raw image data sets need to be converted, typically with compression also being applied, into some accepted standard format of digital image representation, such as JPEG, TIFF, etc. The compression of the raw data inevitably results in some loss of information, and therefore each proposed application needs to be assessed to determine the format of image best suited to its intended purpose.

A frame grabber captures a single image, and is the digital equivalent of a still camera.

Video Capture

A video capture unit captures a series of frames at a specified frame rate. Unlike a frame grabber, which has the luxury of taking time to achieve a better exposure, a video capture unit is required to implement the real-time constraints of the specified frame rate. There are many extremely sophisticated approaches to solving the technical problems, with the only common denominator being the increased cost of the more effective solutions. This is simply because all solutions tend to require computational horsepower, and the more capable / sophisticated solutions require more than the simpler solutions.

The output from a video capture unit is a stream of raw data, which is actually a series of raw frames tied together in a fixed sequence within a video sequence. The data volume is the product of the raw frame size times the frame rate over the elapsed time of the video sequence. Like raw image data, raw video streams are camera specific, and need to be converted, usually with quite heavy compression, into a recognised standard digital video format such as MPEG.

Raw Data Buffer

The raw data buffer is simply the image retention space for images and video sequences captured by the camera and unconverted from the native capture mode. On a consumer camera this is typically quite small, because most storage is done after conversion to a standard digital format. On a FireWire or USB webcam this may actually be only a very few frames, before the raw or converted data stream is pushed down the wire for off-board processing. The steady decrease in the cost of digital storage technologies, and the equally steady increase in capacities, continues to change this equation, but on-board capacities will almost always be orders of magnitude less than off-board data stores.

The raw data buffer becomes important if the frame capture rate can outrun the image conversion and processing systems that require raw data for input. For reasons of speed, raw data buffers also tend to be volatile.

Digital Image Conversion and Compression

The raw image data generated by a camera is always specific to the technology of the camera, which is not generically useful. For an image to be useful it has to be converted into a recognised standard digital image format. Most image formats also offer some compression, although the user is typically able to specify the degree of compression and the loss of information that is acceptable. Most standard digital imagery compression algorithms involve some loss.

The conversion and compression processes are computationally expensive, and large images typically demand significantly greater computing horsepower. Most advanced camera systems use a mixture of hardware and firmware to implement the image conversion and compression functions, with the hardware being based around some form of specialised graphic processor. The graphic processor may be a custom graphic engine or it may be based on one of the customisable silicon technologies such as Digital Signal Processors (DSP), Programmable Integrated Circuits (PIC), Application Specific Integrated Circuits (ASIC), Field Programmable Gate Arrays (FPGA) or other advanced specialised technologies. Software running on a general-purpose computing engine is a very inefficient use of resource, requiring a very powerful processor out of all proportion for the task.

There are numerous digital image formats, the most common being TIFF, JPEG, GIF, PICT, PNG, and so on. There are far fewer digital video standards, with MPEG (**M**oving **P**ictures **E**xperts **G**roup) being far and away the most common, although there are also Microsoft's WMV (**W**indows **M**edia **V**ideo) and AVI (**A**udio **V**ideo **I**nterleaved) and Apple's MOV (QuickTime **Mov**ie) formats in the consumer arena. Up in the realm of professional movie-making cameras, DigiBeta from Sony probably rules the roost. All are supported by a variety of encoding schemes (known as CODECs) designed to reduce the physical size of the recorded video data stream.

Most video compression techniques rely on removing invariant portions from sequences of frames, transmitting a complete base frame followed by only the variant portions from each successive frame. This technique works extremely well when the video source data is taken from a static viewpoint. When a video camera is mounted on a moving vehicle, the motion of the vehicle causes every frame to be significantly different, negating most of the compression potential inherent in the technique of successive frame comparison.

This introduces the problem of image data sizes, particularly when an image data stream needs to be transmitted off-board the originating platform. Commercial consumer digital video averages out to about 10 megabytes of data per minute at industry standard frame rates, at broadcast quality resolutions. By comparison, professional digital video produces rates 25 times higher (250 megabytes per minute) using standard encodings. For real-time transmission of even the most basic such video stream, the data alone will require close to 2 megabits per second, to which must be added the overhead of the transmission protocol (such as 802.11G wireless Ethernet) plus the noise of the network. Several video streams being transmitted in real time could easily saturate an 802.11G network.

Digital Image Store

Images may need to be stored onboard for subsequent retrieval or local analysis, rather than transmitting them off-board. Modern digital storage technologies in the gigabyte range provide significant storage capacities. However, adding any kind of disk into an onboard system introduces a requirement for ruggedization of the storage unit to match the operational environment of the mechatronic platform. Large solid-state storage units are now a practical reality, but at a cost, and they also need to be protected against some environmental factors, such as temperature, vibration, mechanical shock, electrical spike, magnetic pulse, and so on.

The size of the store is not necessarily the most important factor, where there are multiple cameras, the store is going to have to be fast enough to keep up with the continuous data arrival rates from all the cameras. Add to this the notion of some image data also being exported, either to on-board digital image processing applications or via the external communications system to a remote process, and it becomes clear that the speed of the image data store, including the ability to serve multiple clients with continuous near real-time data streams, becomes critical. This is why image-processing workstations typically stripe data in parallel across multiple disks to achieve the necessary throughput. There is no way to push these kinds of data volumes over something like a USB (Universal Serial Bus) or other low speed bus.

Digital Image Processing

Digital image processing is where the rubber meets the road, and there are as many different forms as there are applications for digital imagery. There are two common factors: the first is that all image processing is computationally intensive to the point where all serious applications require dedicated specialised hardware support; the second is that over time a number of semi-standard libraries of image processing primitives have started to emerge. More and more applications are incorporating these standard libraries, although they are standard only in the de facto sense of having proved sufficiently useful that multiple applications have chosen to use them for some base infrastructure support.

The hardware support is a big deal, and is why high-end video cards remain expensive to this day. Image processing is both extremely compute intensive and memory hungry. Most image-processing hardware is based on specialised computing engines rather than general-purpose processors, which introduces the need for application-specific hardware. Image processing engines may be totally custom (proprietary) processor chips, or they may be implemented as firmware running on graphics-processing oriented processors that are designed for manipulating digital signal data. These include the customisable silicon technologies such as Digital Signal Processors (DSP), Programmable Integrated Circuits (PIC), Application Specific Integrated Circuits (ASIC), and Field Programmable Gate Arrays (FPGA) as well as other advanced specialised technologies.

The key point is that each digital image processing application is likely to need to be implemented with its own specialised hardware support. If the image processing is to be mounted onboard the mechatron, it will need to be suitably ruggedized to match the characteristics of the mechatron's specified operational environment. Multiple such applications will need multiple such hardware packages, all of which will require clean electrical power. Digital signal processing hardware tends to produce quite heavy power draws, increasing demand on the typically limited on-board power supply.

A typical image processing application might be scene interpretation, facial recognition, individual tracking, object identification, detection of security violation events and activities, and so on; the possibilities are effectively infinite. Taking security violation detection as an example, there are companies which are entirely devoted to the business of supplying smart automated monitoring of sets of surveillance cameras that can detect classes of security violation and initiate some action on detection, such as paging a human, recording surveillance imagery, or activating alarms and protective processes.

Some applications run off raw imagery, but most will usually accept imagery in a standard digital format. Those applications that work off raw imagery typically have the image capture devices (cameras) built into their hardware package. In other words, they require one or more dedicated cameras. The problem with these becomes mounting the cameras on the mechatron (there may be demanding parameters for placement) and matching the ruggedness to the intended operational environment of the mechatron.

Applications that run off some standard digital imagery format can all share the same digital imagery data store. Doing so requires that (a) the onboard cameras have onboard image conversion support, and (b) the digital imagery store can support the necessary storage and retrieval data rates for all concurrent processes. If new imagery is being stored continuously, and several applications each need to process that imagery in near real time, the imagery storage system must be capable of moving bits in and out at several multiples of the image data arrival rate. If the applications themselves produce modified versions of the original imagery, this too must be stored, adding to the speed and capacity requirements of the image store.

The output from an image processing application is typically a set of objects and possibly a modified version of the original imagery. These objects are comparatively small units of data, and may be textual (for instance, a scene description or the identity of a human) but are more likely to be sets of application specific objects such as 3-D object descriptors. The key is that they are likely to be much smaller than the original imagery from which they are derived, and can therefore be moved around much more readily, even via comparatively low-speed links.

Finally, many image-processing applications need access to reference libraries, which may be too large to be hosted onboard. A library of faces to recognise, for instance, might have to be updated from some remote master database. The issue here is that an onboard image-processing application may require a significant data store in addition to its other dedicated hardware, and that such data stores may effectively be leaf nodes of a remote distributed database.

This brings us back to the base question of "what's it for?" All useful image processing applications are non trivial in the extreme. They are technologies that should be acquired from some organisation that is a specialist in that particular application domain. This means that considerable thought needs to be put into the choice of any application. The corollary is that any company attempting to add camera-based "vision" to a mechatron should not be attempting to create such applications unless they decide to also become a player in one of these image-processing areas, in which case they will need to assemble their own team of in-house experts.

Object Store

The object store is simply a data store for the objects produced out of image-processing applications that are not themselves digital imagery. Ideally there will be one such repository that can be shared by all the onboard image-processing applications, and which can be interrogated or accessed remotely via the external communications link as necessary.

The size and capabilities of the object store will depend on the applications that need to make use of it, but conceptually it will be some form of database management system running as a server, with some form of non-volatile storage medium to hold the data. It will need to be capable of supporting concurrent data exchange with several remote applications, and needs to be considered as a data server on a local area network (LAN) that connects the physically distinct processing platforms for each application.

Control and Data Busses

Cameras are like every other kind of electro-magnetic sensor, only different! The principle difference is the sheer volume of data that a camera produces, and this reflects onto the required characteristics for data transfer. First, all image data transfer paths have to provide very high bandwidth and, second, all data sinks have to be capable of absorbing data arriving at the maximum bit rate indefinitely.

What this means is that any image processing architecture that instantiates the model depicted in Figure 1, or something similar, probably has to be running data paths at 100 megabit or, preferably, gigabit data rates. This really means that the cameras and all associated image-processing and data storage tasks need to be nodes on a high-speed LAN, rather than a more brittle hard-wired backplane. When it comes to external communications, these data rates will easily saturate typical wireless digital networks, especially in noisy environments. For this reason alone, image data downloads should probably wait for a plug-in connection, and video streams that need to be accessed remotely in real-time should probably be analogue (see Section 2).

The data rates of the digital imagery will determine the architecture of the data busses. Typically, control signals and messages will represent comparatively tiny data volumes. However, interleaving control and data messages on the

same bus will almost certainly adversely affect data throughput, particularly when control messages require higher priority. This suggests that there should be a separate bus for control signals, which could also carry low-speed data, leaving the high-speed paths purely for high-volume, near-real-time data streams.

If any of the busses are implemented as local area networks, consideration will have to be given to securing them against eavesdropping and possible foreign interference, whether deliberate intrusion or merely environmental noise.

External Communications

The external communications depicted here are only addressing the requirements inherent in the remote connections to and from an onboard, camera-driven, image gathering and processing system. The thickness of the arrows depicts the comparative data rates and volumes, which are proportional.

External control signals are needed to permit remote intervention in all dynamically controllable systems. It is possible to imagine an autonomous vehicle that is totally self-contained, but in the real world some degree of external control is desirable. Only a stealthy data collection mission fits the totally self-contained autonomous profile.

Downlinking of imagery and the results of onboard image processing are also highly desirable, although ideally part of the onboard processing tasks is isolation of only imagery and objects of interest, eliminating the need for real-time transmission of all image data.

A really major issue for all external communications will be the need for reliable and secure communications. An autonomous mechatron will have programming to cover a loss of signal situation, but it is vital that signals neither be compromised, nor hijacked, nor intercepted. This will necessitate a secure communications system across which both data and control signals and messages can be reliably exchanged. Encryption of the conduit may degrade the available bandwidth, depending on the model chosen, and will require significant processing horsepower at each end to handle the en- and decryption.

Onboard Analogue Video Camera System Overview

Figure 2 gives a somewhat simplified view of the major components involved with an analogue video camera system mounted on a mechatron. Not all the components will necessarily be present in a single camera installation. Typically there will only be a single analogue video camera mounted, although there is no reason why more cannot be deployed.

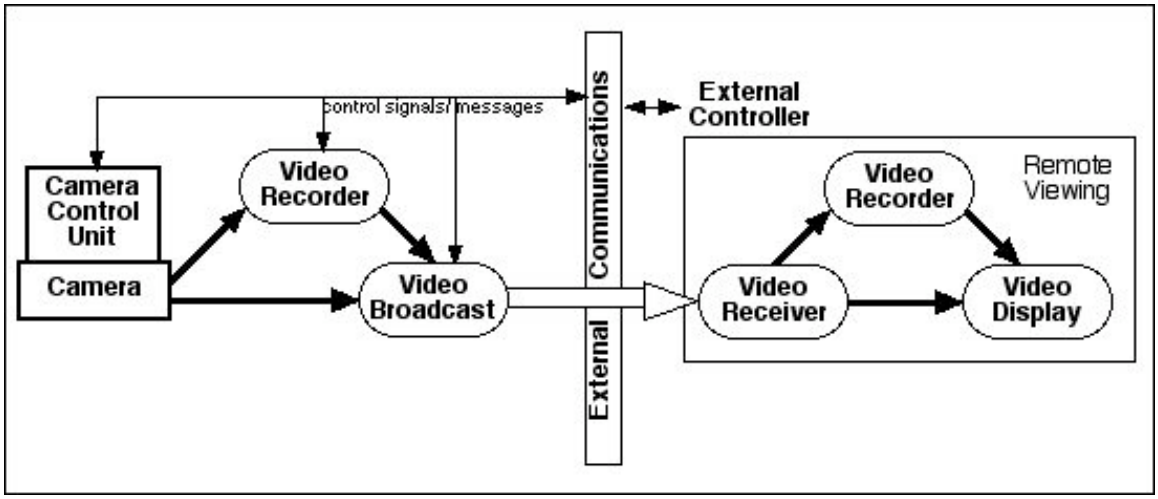

Figure 2: Conceptual Components of an Onboard Analogue Video Camera System.

The major conceptual components of an onboard analogue video camera system are:

1. Analogue Video Camera

2. Camera Control Unit (CCU)
3. Video Recorder
4. Video Broadcast Unit
5. Control and Data Busses
6. External Communications

The following sections describe each of these conceptual components to a level of detail that should be sufficient to enable their relative importance and salient characteristics to be evaluated.

Analogue Video Camera

An analogue video camera is the classic VHS or BETA video camera, which may have no controls at all, or may come with a sophisticated array of lenses, lighting, focus, and exposure controls. Using an analogue video camera eliminates the need for A-to-D conversion to generate digital imagery, and thereby eliminates the possibility of onboard image processing,

The primary advantage is that separate broadcast bandwidth exists for the transmission of analogue video (television) signals, and the technology is well understood, readily available, comparatively cheap, and now heavily miniaturised.

Camera Control Unit (CCU)

Conceptually this is exactly the same as the CCU for a digital camera, although specific functions may differ at the detail level.

Video Recorder

To all intents and purposes this is the equivalent of a VCR, and may even be built into the video camera based on a VHS, BETA, or possibly 8mm tape format. It will need to be connected to a control signal bus to allow remote control, either by a distributed processor onboard the mechatron, or remotely over an external communications link.

Video Broadcast Unit

The primary advantage of using an analogue video camera is that the broadcast and reception of analogue video signals is a mature, cheap, reliable technology, available in extremely small form factors, and, most importantly, with assigned frequencies outside those used by digital networks. This means that the video stream can be broadcast reliably over a completely decoupled, independent link, allowing real-time monitoring of the video at a remote location.

If we want to consider a mechatron-mounted camera that feeds no onboard processing, but which provides decision support and visual monitoring of the mechatron's local field of view to a remotely located human at a control station, this is far and away the simplest, most reliable, and cheapest solution.

Control and Data Busses

The issues are similar to the issues for busses in the digital world, with the high-speed data bus being replaced by analogue video connections. If any of the onboard devices are controllable, then there will need to be both a control bus and some way to interchange control signals and messages with an external controller.

External Communications

The issues have already been introduced in the discussion of camera and VCR controls and the video broadcast unit. External transmission of the video stream will be via an independent analogue TV broadcast unit. Command and control signals will need to go through an external communications link, conceptually identical to the one discussed in the communications section for digital cameras.

Single Camera Issues

Single camera installations trade mechanical simplicity for conceptually less capability. However, the processing required to automatically fuse information derived from multiple discrete sets of imagery is incredibly complex. This is true even when the cameras are co-located in known relative positions on a single platform, and the images are coordinated with timestamps and other key identification data. Processing imagery from a single camera severely restricts the computational envelope, rendering it somewhat more tractable.

The single camera mounting is the canonical case, where the intended use effectively defines all the selection, placement, and mounting criteria.

When only a single camera is available, it becomes necessary to define the mission parameters in order to select the best mounting. For instance, if 360-degree coverage is required, the camera must be mounted high in a mount that permits unrestricted rotation. At the other end of the scale, a fixed camera intended solely as a navigational aid might be in the middle of a front panel, looking forward and slightly down. The intended application will dictate the optimum position and type of mounting, as well as the type of camera.

For example, if the camera is intended to recognise human faces, it probably needs to be mounted at the approximate height of a human face. If the recognition algorithm demands a direct look at the face, the mounting probably needs to be height adjusting to match the varying heights of human, including seated versus standing, and other such idiosyncrasies of human behaviour. Alternatively, the mounting needs to have a mechanism that can persuade humans to look directly into the camera. However, this latter approach then rules out facial recognition of the uncooperative, paralysed, trapped, unconscious or dead, any or all of which may be unacceptable.

As a final aside, the reason that most robot-mounted cameras are mounted high is simply because it increases the utility of the camera. The higher the camera is mounted, the greater the resultant field of vision. Plus it is easier to obtain wider lateral coverage from a top mount. If humans had eyes mounted in their chests, they would have to turn the whole body, not just the head, to see to either side. Think of a chameleon for the ultimate specialisation in flexible vision.

Multiple Camera Issues

Multiple cameras do not automatically imply attempting to mimic human binocular vision. The physics of cameras is so different from the mechanisms of the human eye that it isn't a practically attainable goal anyway. Having multiple cameras simply allows the concurrent support of more than one image processing application, without necessarily having any fusion between the multiple image sets.

Many image-processing applications have different input image requirements, and dedicating separate image-capture devices (typically cameras) to each process is often the best way of satisfying them. This does not rule out fusion of the image data from multiple cameras, but this requires an application specifically designed to perform this task. The most probable multiple-camera scenario has separate cameras for navigational (piloting) assistance and for surveillance. The choice of several fixed direction cameras versus one rotating camera becomes a technical choice, possibly including considerations of cost and mechanical complexity, but most likely based on intervals between imagery update along a given axis.

Software that attempts to fuse concurrent imagery from multiple cameras is significantly more complex than that needed to perform the same application using only a single set of imagery. This is in part because it is very, very hard to distinguish between two or more different views of the same object from views of several different but possibly similar objects. This is true even when the geometry between the cameras is fixed, and all the operational parameters for each camera and each image frame are known. This is just a very hard problem, for which research has as yet failed to yield elegant or even general solutions. All the extant applications have a very limited and equally specialised focus.

Adding a second camera, or even several cameras, is only useful if each additional camera supports an application, an application that in turn justifies its inclusion on the robot vehicle.

Onboard versus Off-Board Processing

Onboard image processing requires adding substantial, and possibly power hungry, specialised computational power to the robot vehicle. Performing the image processing externally allows these computational resources to be centralised and optimised, at the cost of having to telemeter the image data from the vehicle to the central processing system, and return the results.

The drawbacks all lie in the telemetry, because limited bandwidth makes it very hard to transmit continuous image data streams in real-time. This is true under ideal conditions, and exacerbated when conditions are less than ideal, which is typical in outdoor, wide area operations. The system is also totally reliant on the communications working in a reliable fashion, with the system losing all capability when the link degrades or is lost.

The problem with performing the image processing on board is that all the image-processing packages of today are extremely resource hungry. This results in extra cost, a heavier power draw, and the need to protect additional, comparatively fragile processing hardware from environmental damage. The gain is in greater autonomy with less reliance on reliable communications to a remote station, plus the elimination of very high volume data transmissions. These latter could, in extreme cases, assist an enemy locate a hidden observing vehicle, even if the data content is encrypted. Secure, high bandwidth, data communication channels are themselves non-trivial items.

Onboard Cameras as Navigational Aids

There are two ways to use one or more onboard cameras to assist in vehicle navigation. Here the term navigation probably more properly refers to piloting or driving, meaning the local stance rather than the global stance. Cameras are by definition limited to line of sight, so their principal use is in detecting and recognising objects, which may present obstacles to movement, or may act as landmarks.

The simplest possible model is a fixed, forward facing camera that transmits images (either single frame or video) over a link to a remote operator's console, where it can be displayed to assist the remote operator control the vehicle. Again, the simplest model has the remote operator assuming direct control of the vehicle via a tele-remote control set.

One step up from tele-remote operation is a semi-autonomous vehicle with assistance from a remote operator. Essentially, in this model the remote operator examines the up-linked imagery for landmarks and obstacles and has some mechanism to return the object recognition and identification information to the vehicle. This requires a fairly sophisticated interactive display that the operator can use to identify and classify or otherwise annotate objects appearing in the field of view. Clearly, the faster the vehicle is travelling, the less practical this approach becomes, because of its indirect nature. The process involves the following sequence of steps:

- Camera imagery up-linked from vehicle to remote control station.
- Camera imagery displayed to remote operator.
- Remote operator detects an object in the displayed imagery.
- Remote operator classifies and possibly annotates object in display.
- Remote control station downloads object information to vehicle.
- Vehicle navigation system interprets information and acts accordingly.

The mechanism via which the remote operator classifies and annotates objects appearing in the camera's up-linked imagery is likely to be the most time-consuming part of the process. Even if the operator has a specialised pointing device with dedicated buttons to trigger pre-programmed functions such as "lasso select" and designate as obstacle, the probable response time between the remote operator seeing an object in the display and the identification information being transmitted back to the vehicle is likely to be several seconds at best. When there are multiple objects, the remote operator will have to perform additional steps to decide on a priority order for classifying all the objects in the field of view, or at least all the objects that may be important to vehicle navigation.

The next level up is to automate the process of object detection and recognition. This process may be based solely on the camera-generated imagery, or it may include fusion of data from other sensors to help determine the location and characteristics of detected objects. This automation can run remotely, in which case the camera-generated imagery needs to be transmitted in real-time to the remote analysis system. Alternatively, the image processing can be performed on board the vehicle, in which case a suitable image processing system (hardware and software) must be built into the vehicle.

As with the remote human operator, the raw processing horsepower required to keep up with the vehicle increases with the speed of the vehicle and with the number of objects appearing in the camera's field of view that need to be classified. Consider that a vehicle travelling at 60 km/h is travelling at just less than 17 metres per second. If the camera's visual detection range is, say, 50 metres, this would mean that there is only a 3-second window in which to not only perform all the necessary processing, but also to complete whatever mechanical reaction is needed for the vehicle to avoid, stop, slow down, or otherwise adjust its trajectory to the presence and nature of the observed object.

It is worth noting that the latest generation of SUVs and minivans comes with a downward pointing, rear facing video camera linked to a dedicated dashboard display to assist drivers in reversing manoeuvres. Using onboard cameras for autonomous vehicle navigation requires an image processing system capable of replicating a human driver, albeit without having to add the layer of understanding that says that a detected obstacle is (say) a lamppost or a hydrant. It is sufficient to detect a solid, narrow, vertical obstruction in the vehicle's current path.

Note that the image processing system identifies the object and is responsible for classifying it as an obstacle to navigation with a precise location and dimensions. It is up to the vehicle (auto-) pilot to decide whether the obstacle is in the vehicle's current path and, if it is, what actions to take. The possible actions will depend on the nature of the vehicle, and range from avoidance to stopping to destroying the object at a distance to deliberately hitting the object; the latter possibly because the vehicle is sufficiently robust to either destroy the object or not be adversely harmed by the contact. Piloting requires both mapping and path (trajectory) plotting capabilities, each extremely large research domains in their own right.

The process of identification and classification of objects in imagery has been the subject of research for decades. Huge increases in cheap computing power, and in specialised graphics and image processing engines, as well as the arrival of low cost cameras with excellent resolution, have all contributed to an upsurge in the quantity of the research being done. Quantity, however, does not necessarily contribute to quality, and there have been few if any quantum leaps or major breakthroughs in the field.

A Canadian company, Deep Vision Inc. of Dartmouth, Nova Scotia, claim a major breakthrough in this area, but the material they make publicly available fails to convince that it is either a quantum leap or directly usable without a very large effort to build up dictionaries of described objects.

Earliest image analysis focused on detecting edges, and good edge detection allows for finding and following a well-defined path. There are hobby robots (such as the Cybot, originally from Durham University) that also perform edge detection and can follow a line with high enough contrast between the line and its surroundings, but do so using an incredibly simple vision system dedicated to line following. This is typically by means of a little optical emitter-detector pair, in effect acting as a very simple fixed focus camera. When the edges of the path are not well defined, and when the visible scene is complicated by objects with much more clearly defined edges (buildings, poles, tree trunks, etc.), the problem becomes hugely more complicated.

The next level up searches for static solid objects, then moving objects, and also considers amorphous objects such as leafy bushes, tall grass, and so on. These are all complex problems with no reliable, low cost solutions. The good news is that the emergence of consumer digital cameras has resulted in standards for the representation of digital imagery. This in turn has caused a number of research efforts to produce comparatively self-contained libraries of useful primitive operations that work against standard formats of digital imagery.

The other good news is that, while most academic and research institutions work on UNIX platforms, Microsoft Windows has become so ubiquitous, and therefore such a large potential target, that most such libraries are also available with a version that will run on a Windows platform.

Even given a camera package generating imagery in a standard format, and a library of standard scene interpretation and object identification primitives, creating a navigation application is still a non-trivial task. Coupling such a navigation capability to an autopilot is a further non-trivial step. Add fusion of the data from other sensors into the equation and the problems become still larger.

In conclusion, it is extraordinarily difficult to create a purely camera-based vehicle autopilot, particularly if vehicle or environmental safety (keeping objects in the environment safe from vehicle impact) considerations are critical. It is incredibly difficult for a human to decide whether a vertical surface blocking progress is a wall, a curtain, or a sheet of paper. Collision with the wall will damage the vehicle, with the curtain will entangle the vehicle, possibly damaging the curtain, and with the sheet of paper will destroy the paper and probably not cause the vehicle any problems. It is probably better to try and detect the nature of such an obstacle via other sensors.

Note that highly successful commercial mobile robots such as iRobot's Roomba do not use cameras at all. This suggests that vehicle navigation should probably only use camera-derived imagery to augment a system that is primarily based on other kinds of sensors.

Vision and Computational Complexity

There is a real question as to what exactly the term vision means. This is because cameras do not behave like biological eyes, nor do we have any deep understanding of the cognitive processes associated with biological vision. It is best to think of a camera as simply a sensor operating in the visible spectrum, and capable of yielding continuous sequences of snapshots (frames) at regular intervals. The intervals are usually set as a commanded intention, which may be modified by physical imperatives and conditions of the operational environment.

Computer (or robot) vision then simply becomes the applications that can be defined to process sets of camera-created imagery. If this processing is performed in real-time it can be more readily equated to biological vision, although in practice there is no conceptual computational difference between real-time and non-real-time processing beyond the temporal constraints. Certainly, if an image-processing application can analyse image data in real-time, there is no reason why it cannot also run against stored imagery at a later time.

This introduces the need to distinguish between real-time expressed as a rate, meaning that the process can keep up indefinitely with the rate at which source data arrives, versus wall time, where the process is required to synchronize with a so-called wall clock that displays the real time, e.g. the signal from a standard time server. In simulations the wall clock rate can be stepped up (a so-called fast clock) to compress event sequences, or it can be stepped down to give slow-motion behaviour.

Because image source data is very rich, the data volumes are very large, in consequence of which all image processing is both memory and processing horsepower intensive. If each frame contains megabytes of data, it takes time to process every unit cell of a frame (typically known as a pixel, for picture element). When processing needs to span multiple frames, both in-memory storage and computational horsepower requirements rise, often in exponential rather than linear fashions. General purpose processors make poor platforms for image processing because they are non-optimal over the limited range of appropriate processing options, and too sluggish when necessary primitive operations need to be constructed as complex functions from the general-purpose instruction set. All serious image processing is committed to specialised processing engines that have been optimised for performing the typical low-level functions fast.

Such processors have sophisticated multi-port memory models, and often support pipelining to improve the throughput, because most image processing involves a great many repetitive routines that are amenable to heavy parallelization across multiple data elements. In fact, most of these specialised processor architectures are also designed to scale up or extend via ganging multiple processing units.

At present, most image analysis approaches are layered, starting with conceptually simple processes that detect fundamental objects, such as edges or nodes or colours, and output an overlay map with these objects plotted. Once the fundamental objects have been catalogued, the next stage is to start bundling groups of objects according to associative

or other filtering rules to identify complex objects, which include polygonal facets, and allow for partial obscuration, lighting, reflection, translucence, and all sorts of other light-related physical phenomena of camera-derived imagery. There can be many recursive layers of such processing before a final object map is generated.

The objects in the final object are those which a human would see and attempt to identify, and it is at this stage that image processing moves away from the purely mechanical (the application of physics and mathematics) and into the realms of cognition, reasoning, and other knowledge-based processing techniques. Alas, at the present state of the art, all of these techniques are also computationally intensive (expensive), and may involve attempting to match against very extensive dictionaries that describe known real-world objects in a form that is amenable to pattern matching. This is the computational area where Deep Vision, Inc. enters the picture, so to speak.

In common with a number of other research disciplines, a vast amount of this processing still runs on a largely trial and error basis, resulting in significant processing overhead. This is why many computer vision systems are designed to be trained and, once trained, can run in a much faster closed world mode. Training often involves technologies such as neural nets, and works very well for closed environments such as product assembly lines. Changes to an assembly line are always planned, tend to take significant time to implement, and therefore provide a clearly defined window for retraining. This doesn't really work for a general-purpose surveillance robot operating in the open, where there is a high probability of encountering a never previously seen object.

An enormous amount of work has already gone into attempting to recognise common natural and artificial (human constructed) objects, but the problem space is extremely large, especially when the objects are small (or physically large, but at a distance, and therefore small in the imagery), partially obscured, or, worst of all, moving. Consider the problem of identifying a strange bird from a book of common birds. At best you have several full colour images of different aspects of each bird, plus a detailed textual description of the bird's physical characteristics. You also probably have a description of preferred range and habitat, which can increase or decrease the likelihood of a candidate. However, on February 5th 2006 security at Dulles international airport (outside Washington DC) was severely disrupted when bird-watching enthusiasts turned up in droves on the rumour of a unique sighting of a snowy owl, completely outside any historically observed range.

How would a vision-based observational system determine if a distant flying object were an aeroplane (large at a distance, or small and nearer), powered or unpowered, a bird, a helicopter, a lighter-than-air craft, or an unidentified flying object? Even humans have trouble with this, and rely on experiential interpretation of trajectory and flight behaviours over time. Sometimes it is a feat to even locate a small flying object when someone else is trying to point it out to you! A whole new set of skills needs to be deployed when the problem is recognition of a camouflaged animal in cover, whether stationary or moving. Worse, detecting a stationary animal is a completely different problem to detecting one that is moving. And so it goes. Each recognition problem is different, relies on surprisingly deep knowledge and cognitive interpretation, and there are an almost infinite number of such problems; the trick is to pare down the set of such problems to a minimum that apply for a given application.

These examples are simply presented to try and provide some perspective on the enormous complexity involved in extracting real world information from a stream of image data, whether from single or multiple cameras (viewpoints).

Potential Applications of Onboard Cameras

The range of potential applications for camera-derived imagery is almost as broad as the range of human endeavour. Even limiting this to probable applications for cameras mounted on a mobile robot vehicle still leaves a staggering array of possibilities.

Who's Out There Doing What

For anyone wanting to make a quick survey of the technologies, product offerings, and solutions that are available, here is a small list of commercial companies that are to some degree representative. This list is only meant to provide a sample of typical offerings, and is neither an exhaustive catalogue nor a complete survey of any technology.

- **Amerinex Applied Imaging** – The Imaging Understanding Company
  <http://www.amerineximaging.com/index.html>
  Amerinex Applied Imaging, Inc., specializes in computer vision products, engineering, and research. Since 1995, Amerinex Applied Imaging, Inc. and ADCIS S.A. in France have been jointly working on the development of the Aphelion™ imaging software product, a Windows™ based product to quickly develop and deploy imaging solutions.
- **Cognex Corporation**
  <http://www.cognex.com/>
  Cognex Corporation is the world's leading provider of vision systems, vision software, vision sensors and surface inspection systems used in manufacturing automation. Cognex is also a leader in industrial ID readers.
- **Coreco** (DALSA Corporation, machine Vision)
  <http://www.coreco.com/>
  DALSA Corporation is an international technology leader in the design, development, and manufacture of digital imaging products and solutions.
- **Deep Vision**
  <http://www.deepvision.ca/>
  Deep Vision specialises in developing real-time, intelligent machine perception technology. Deep Vision has developed a unique and proprietary technology that enables systems or devices to detect, recognise, and interpret objects and events, in their environment, with blazing speed.
- **Evolution Robotics** (sic)
  <http://www.evolution.com/>
  Evolution Robotics are a leading provider of a Robotic Operating System with modular solutions for Autonomous Navigation, Man-Machine Interaction and Integration available through licensing or as part of Development Kits such as their ViPR®, self-described as the gold standard for visual pattern recognition.
- **Mango DSP** – Intelligent Video Servers
  <http://www.mangodsp.com/>
  Mango DSP is a developer and manufacturer of intelligent video surveillance devices. Mango's encoder, DVR and IP Camera technology powers many of the OEM solutions sold by fortune 500 companies in the Homeland Security, Defense, Retail, Mobile and transportation markets.
- **MVTec Software GmbH** - Machine Vision Technologies
  <http://www.mvtec.com/>
  MVTec is a leading international supplier of standard software products for machine vision. Our industrially proven software products HALCON and ActivVisionTools are used in a large variety of application areas, such as semiconductor industry, inspection applications, medicine, and surveillance.
- Neptec
  <http://www.neptec.com/>
  Neptec is the Leading Supplier and Integrator of Machine Vision Systems for Space Applications. Pioneering the area of intelligent 3D vision, Neptec designs and builds smart sensor systems capable of extracting, classifying, identifying and tracking objects while enabling autonomous navigation functions such as obstacle avoidance and path planning.
- **Neven Vision** – acquired by Google in August of 2006
  <http://www.nevenvision.com/>
  Neven Vision is a key player in face and image recognition biometrics, with deep technology and expertise around automatically extracting information from a photo. It could be as simple as detecting whether or not a photo contains a person, or, one day, as complex as recognizing people, places, and objects.

- NRC – National Research Council Canada
  <http://iit-iti.nrc-cnrc.gc.ca/>
  The National Research Council of Canada has a long history of research and innovation in the domain of machine vision. Use their web site search functions to narrow down the field to the specific disciplines in which you are interested.
- **Sightech** – Machine Vision Systems
  <http://www.sightech.com/>
  Sightech produces machine vision systems for quality inspection, industrial automation, automated sorting, defect detection, and process monitoring applications. Sightech is the leading supplier of intelligent self-learning inspection systems.

The generic name for this whole subject domain is machine vision, and searching on the web using this term plus one or more focus search terms to narrow the field to a particular discipline will provide a gateway into the huge body of research, technology, and even products and services that are available.

Produced by: Robert Stanley, Principal, Blue Rabbit Consulting Inc.
Dated: February, 2006

Blue Rabbit is a consulting company in Ottawa, Canada dedicated to creating winning strategies for technology-based companies.
Telephone: (613) 692-3868 or rstanley@bluerabbit.ca